Feature request: Greenshift CSS Builder & Manager

v1.0

Jump directly to Feature Request Proposal

Greenshift has a great UI for styling blocks, making design fun and easy.

But currently, the only way to have styles not limited to the current page is either:

- create global classes/variables
- add some global raw CSS.

In one case there are some limitations, it’s not easy to manage and it loads up the database, especially on heavy usage of global classes/variables (it is said maximum 50 classes), or it’s not very convenient with raw CSS...

Limitations (non exhaustive)

1. Use GS styling UI to style any valid css selector, and not only current block or its children

Example: set some default styling for any basic elements like "section", or link "a" tag in the page
The only way for now is with raw global css, or by creating an element that wraps all other blocks (so you can style them via local class subselector), but it’s limited

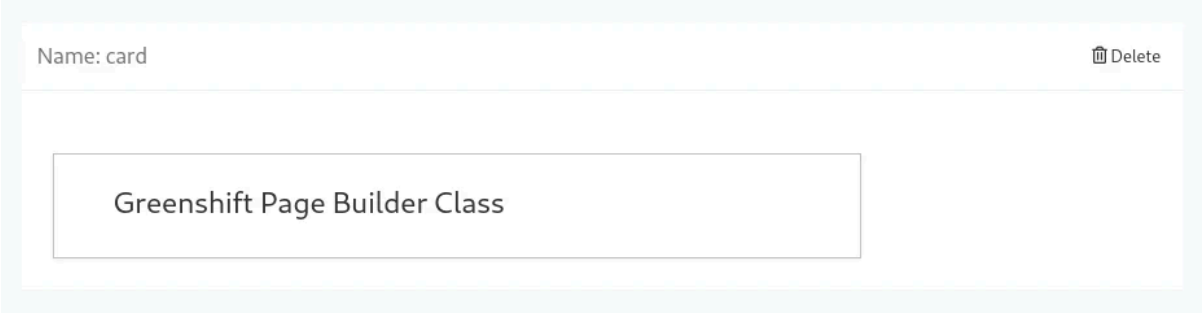
Example: you can’t style some link "a" tag from classic theme, say in the nav header, because you would need to wrap it in a GS/GL block, add some local/global class to the wrapper, and add a `a` selector to target it.

2. Have a centralized and visual way of building/managing classes/styling

Take this card design for example:

Delicious Desserts
SEASONAL INGREDIENTS
Vel nam odio dolorem, voluptas sequi minus quo tempore, animi est quia earum maxime. Reiciendis quae repellat, modi non, veniam.

You won't design a `.card` global class like this and it's subselectors inside Stylebook global classes UI, that shows only a div with some text in it.



You need to design it with its internal layout and content examples, so for now you have to find a page somewhere where this class is used and then style it, but it's not convenient.

3. Make variations of a class (or any selector) based on ancestor elements

Example : add an animation effect on a `.card` class when body element has a certain class. we have no way to set `body.xxx .card` styling except with raw css, and thus, no GS styling UI like keyframe builder to do this.

4. Organize your classes

Currently in global classes in Stylebook, it just show the list of all your global classes. You can’t group/organize/arrange them like you would want to, or show only some part of it if you have many classes.
Same for export, it’s all classes at once

5. Easily move around our styles/classes

Imagine a website with 50 global classes, I only want to pick 5 of them (pretty complex ones, many subselectors etc) and add them to another site, I can't really do this

6. Versioning of global classes

If you do a bad change, you can't really come back easily

7. Rename global classes

Currently, rename button creates a new global class

8. Working with GS global classes can be misleading

Example: You want to set some color and other stylings on a paragraph.
As you may reuse this design on other pages, you make it a global class, and use it through many pages.

Now you want to use this class on link elements also (or any other tag), but there is a problem:
This global class is outputting this CSS:

```
body .global-class:is(p) {
  color:#f11414;
}
```

Now you’re stuck to only use this class on paragraphs only.
It's still possible to fix this though, without having to make a new class and change all the classes everywhere, but it is very tedious, and you can’t do this in stylebook directly (as there’s no brush icon to copy styles there):

1. Find one of the paragraphs with your `.global-class`
2. Add a `div` element
3. Add a `global-class-backup` **local** class to the div
4. Come back to the paragraph, click on the `.global-class` > Brush icon > Copy styles
5. Go to the div > `.global-class-backup` > Brush icon > Paste styles > Add style attributes and selectors (if any)
6. On the paragraph, click `.global-class` then red Trash icon (you can make sure that the deletion worked by looking in stylebook)

7. On the div, create a new `.global-class` (by doing this on a div, there will not be the `:is(p)` thing)
8. Copy styles from `global-class-backup` and paste to `.global-class`, then **save** the global class
9. Re-add the `.global-class` on the previous `div`, remove the div and save your page
10. Now you can use your class on any type of element

This seems to work, but I don't even know if it is a reliable way

You also get a CSS rule with less specificity, depending of what you're trying to do, it can be better or not:

In any case, you can't tune it your way, this is another limitation

Feature Request Proposal

I thought of some solutions that would require the least amount of work to implement by reusing as much as possible existing features and ecosystem of Greenshift and WordPress, but also being very versatile as you can use it in many types of workflows.

This way, it tries to not introduce any fancy/difficult new things to grasp. I may say that 99% of the features already exist! It’s mainly a matter of how to arrange them in 2 new wrappers:

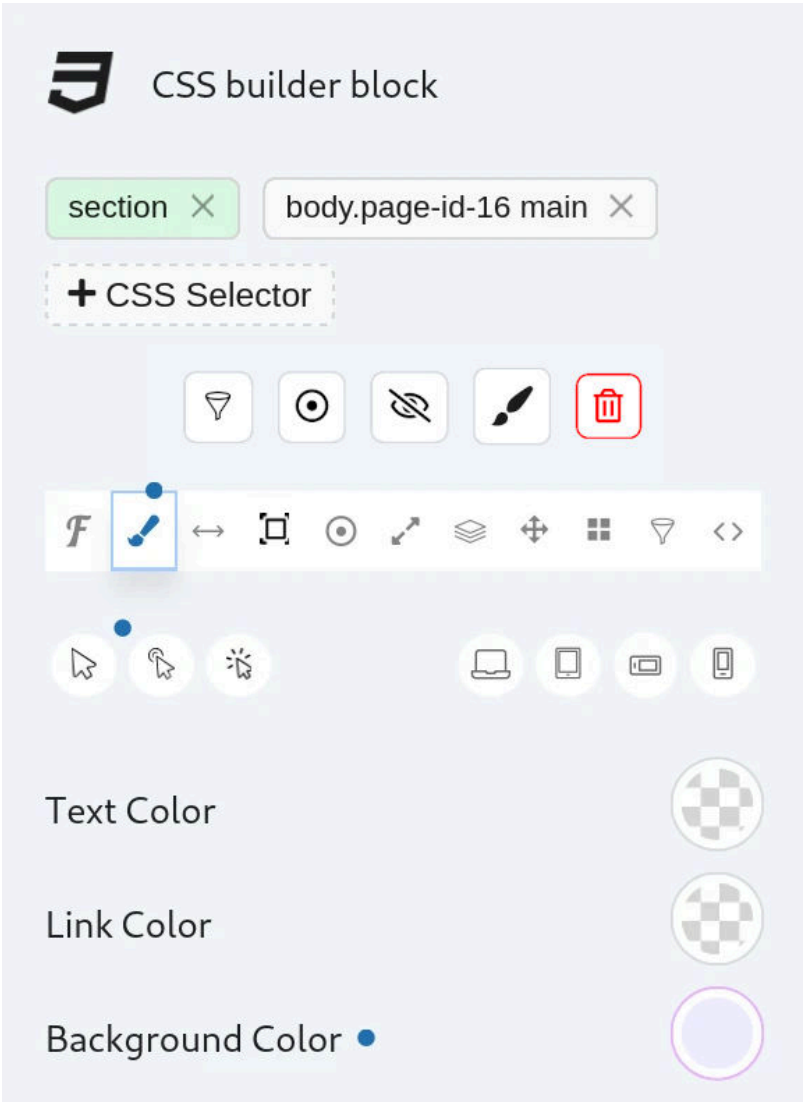
Introducing the new CSS Builder block

The base of this proposal is the new CSS Builder block:

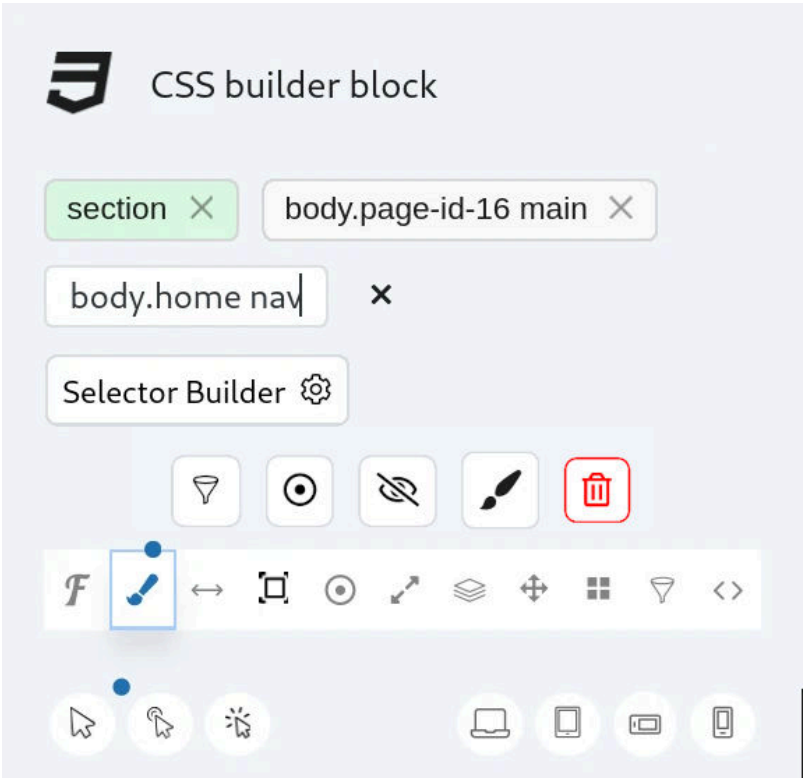
It is "very" simple: It’s just a block in which you can add any type and amount of CSS selectors, and style them the same way you do for local/global classes.

It does not show anything in the editor.
It can be added alone or you can add any other blocks inside it, it will just output all the designs you made in it, in a `<style>` tag.
Any block inside it will just render as usual after this tag, it’s just a possibility for how you can organize your blocks as you’ll see later.

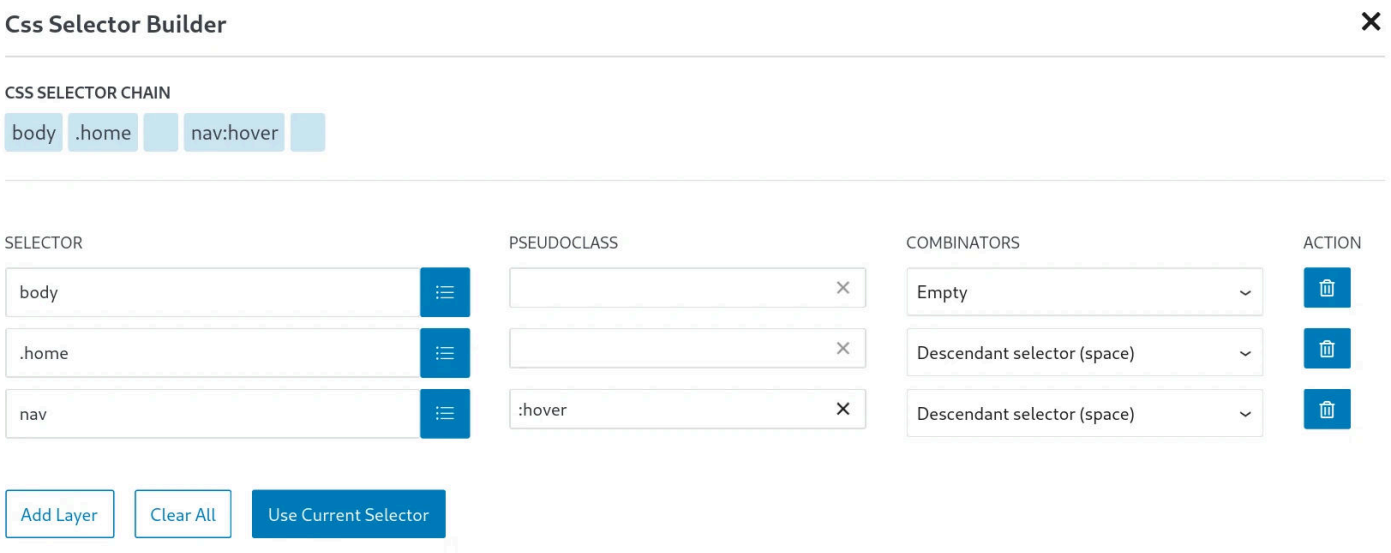
Here is an example for adding a background color to all `section`s by default, and some other styling for a specific page:



You can add other selectors in the same CSS Builder block, or separate your work among multiple blocks, it’s as you wish.
You add them the same way as you do currently with subclasses, either by typing your selector by hand:



Or by using the selector builder:



This block let you design and target virtually anything you want where it is rendered, so not only what is inside your content, which is the current state in GS.
In this example : *target the styling of the* `nav` *element, when it is hovered, only on WordPress home page*

Also, if you just create a simple `.something` class, you will find it as usual as a classic class in the class selector of GS blocks.
`.something.other-thing` would create two classes

Subfeature request : we need auto classes and create selectors buttons in brush icon of classic classes

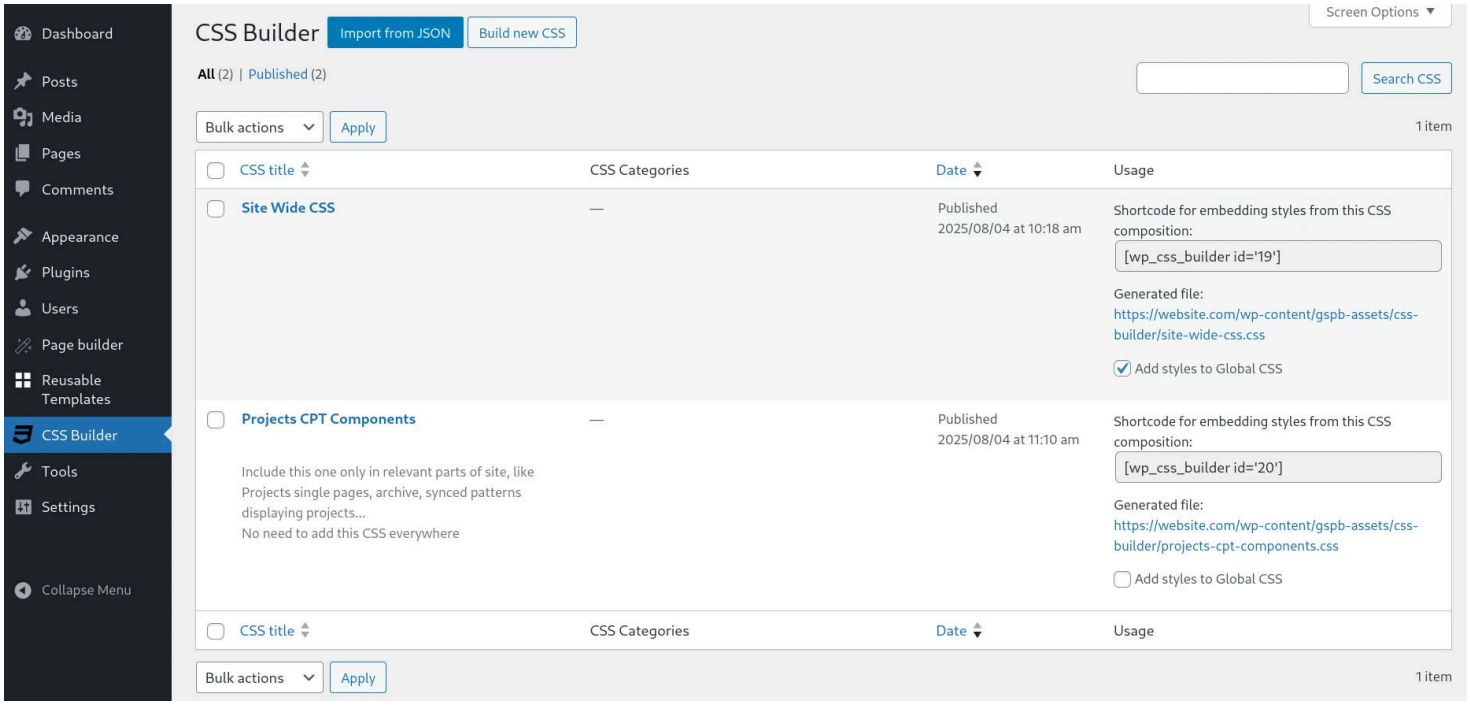
---

The CSS Builder block is already useful by it's own. By leveraging core WordPress features, you can create many workflows:
You can add it to single pages, posts, [synced] templates (you even obtain a shortcode to embed with this), FSE patterns...

But with a little addition, it can become even better:

CSS Manager

Introducing the new CSS Builder manager menu:



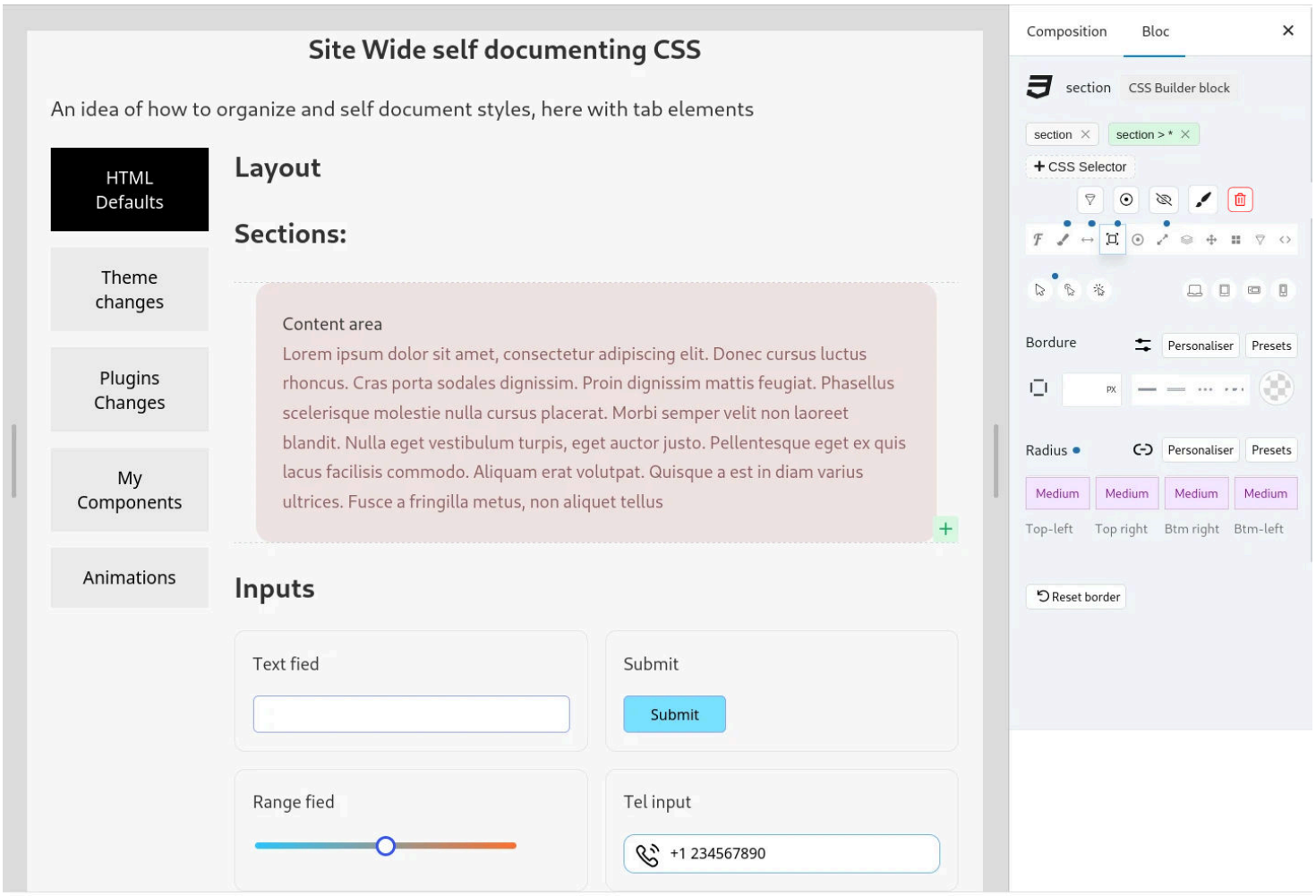
It's virtually the same thing as the Reusable Templates CPT menu:
You can create and manage CSS Compositions; export/import them, either here by json or by copying any blocks you want inside the editor.

The slight difference with Reusable Templates is that:

- They generate a css file from stylings made in all CSS Builder blocks you added in there, that you can download/link to.
- There is an option to automatically add the styles to the global css on frontend
- When you embed them (in posts, pages etc), it only enqueue those styles, any other blocks inside them will be ignored
- It creates a page where you can see the rendered version of what is in editor (so with all the blocks this time)
  Needed to check the actual rendering on frontend, as editor can be weird sometimes
  And useful to navigate your designs

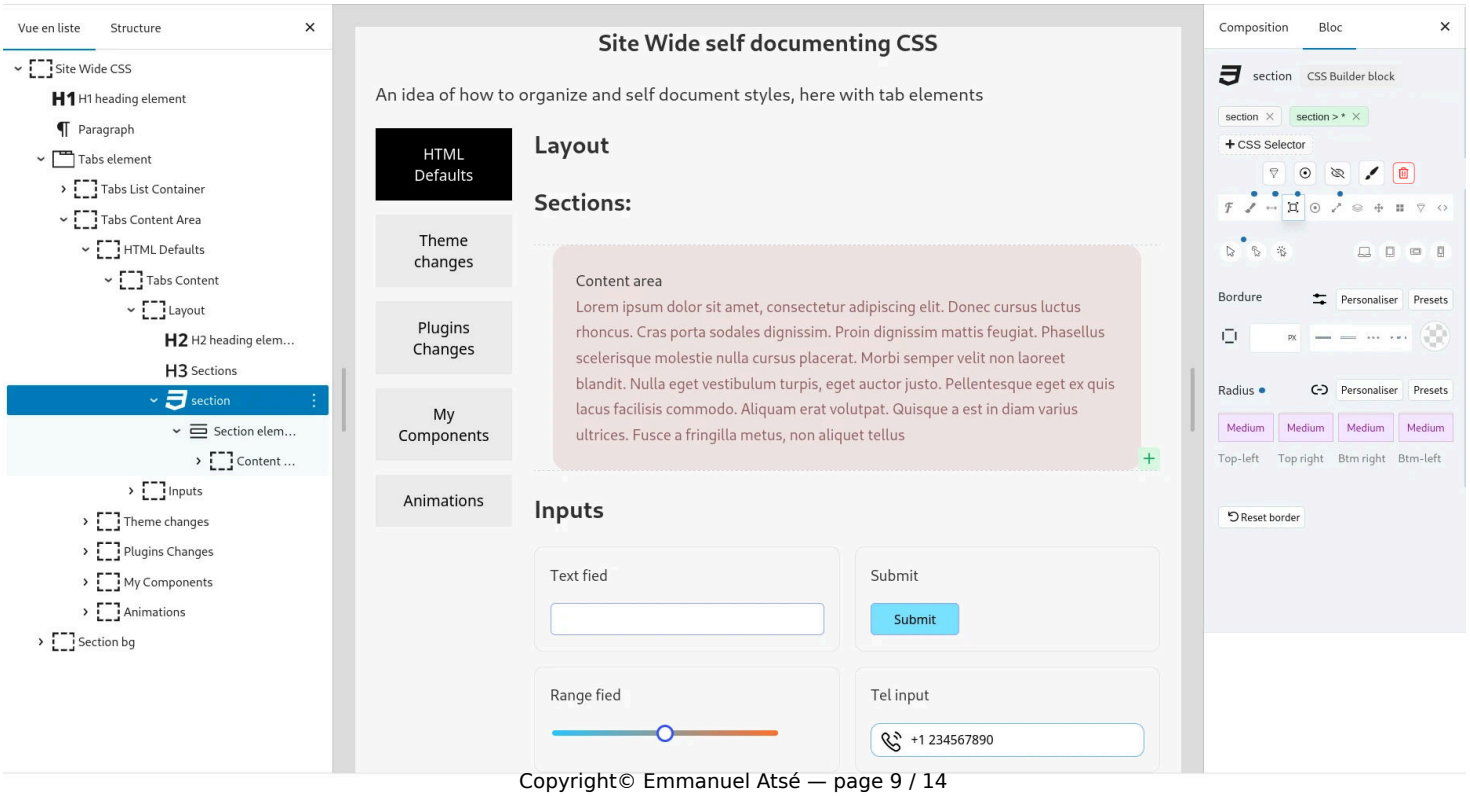
That means... that you can build your own self documenting and visual CSS/Design! (and automatically versioned by WordPress)

Here is an example idea:



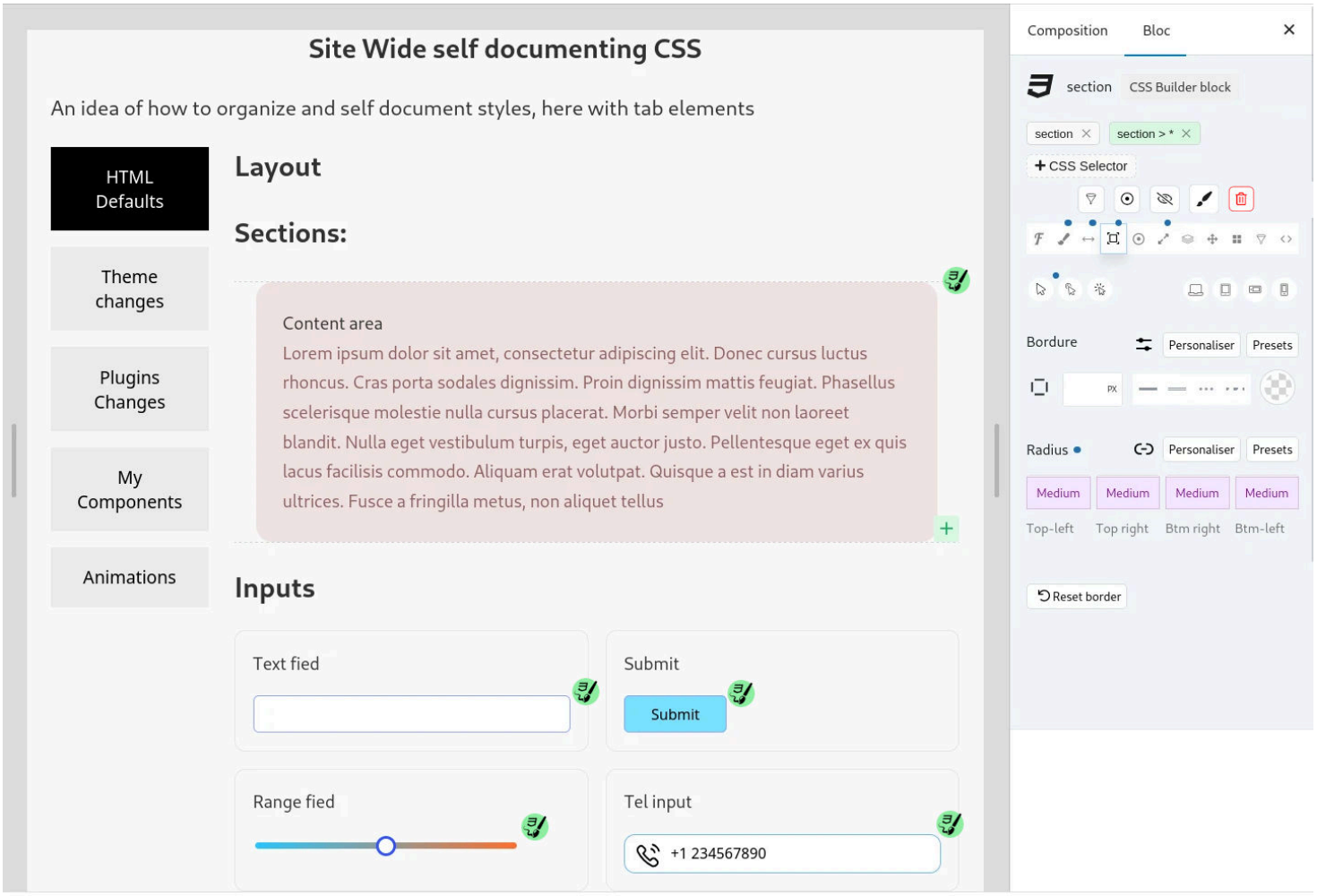
I got this idea for organizing some default general styles for all the website
You're totally free to do this the way you want; you could just throw a bunch of CSS Builder blocks with absolutely no other blocks (and thus have just a blank page), it would work the same way.
Or make a simple list; document how to use certain complex classes; show variations... The [Greenlight Pro Brandbook] is also an example of what you could do

Let's see how this is organized:



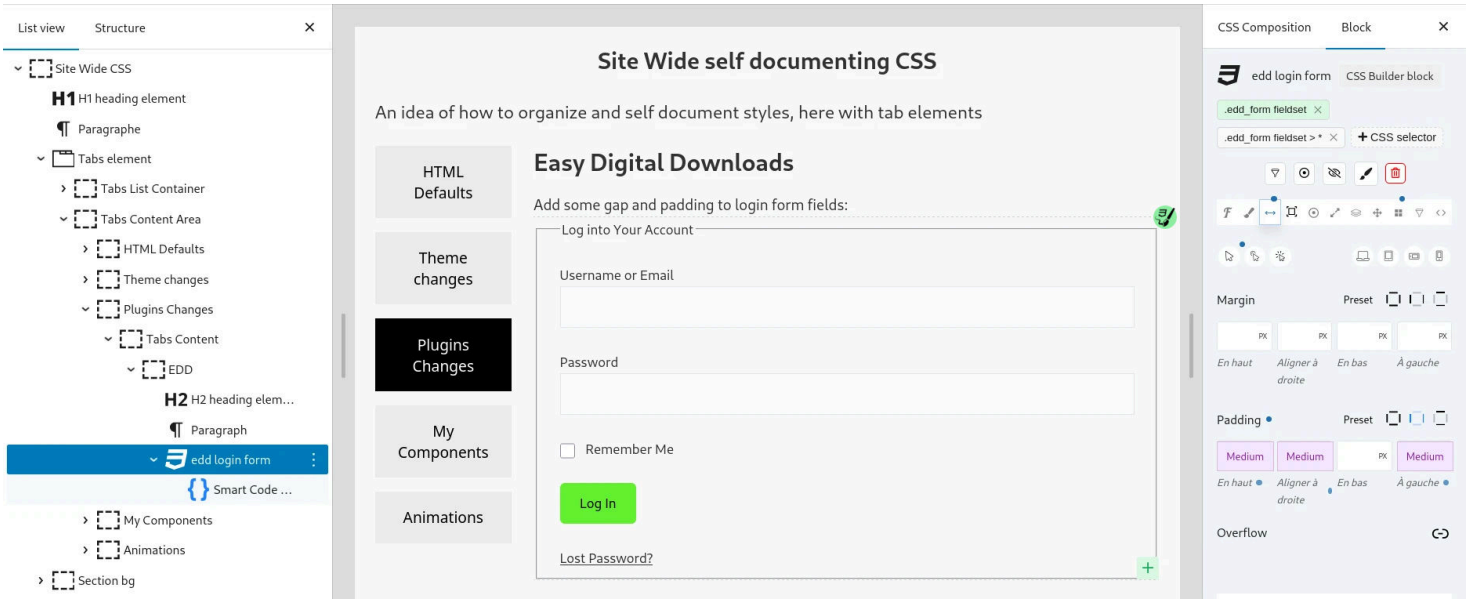
It's a classic design, here with tabs element. I just wrapped relevant parts with CSS Builder block so it's easy to find.
In CSS Builder block, you see the styling menu for section content area (`section > *` selector).

But a visual option makes this even easier:

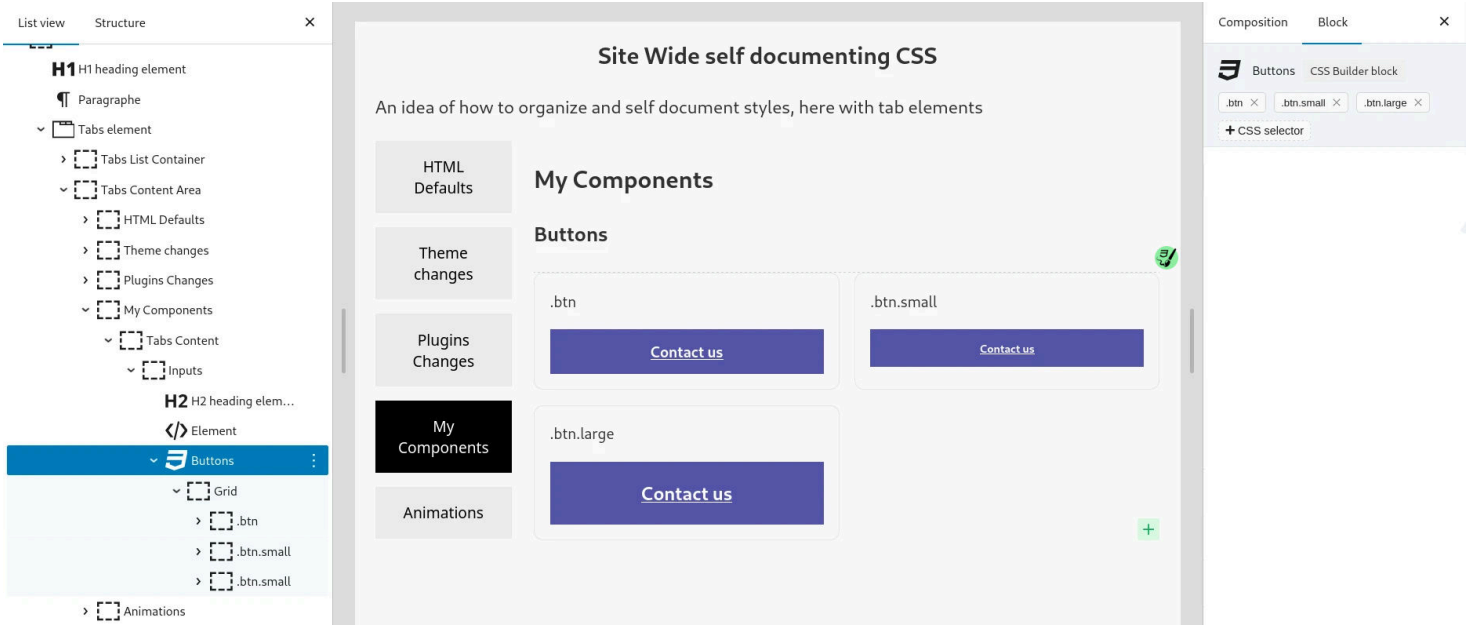


As CSS Builder blocks are invisible (so you would have some clicks to do to get to them), It displays a little quick edit buttons on wrapped elements that make you select them directly.
It's also a nice indicator of which styles are relevant.

Some other tabs:

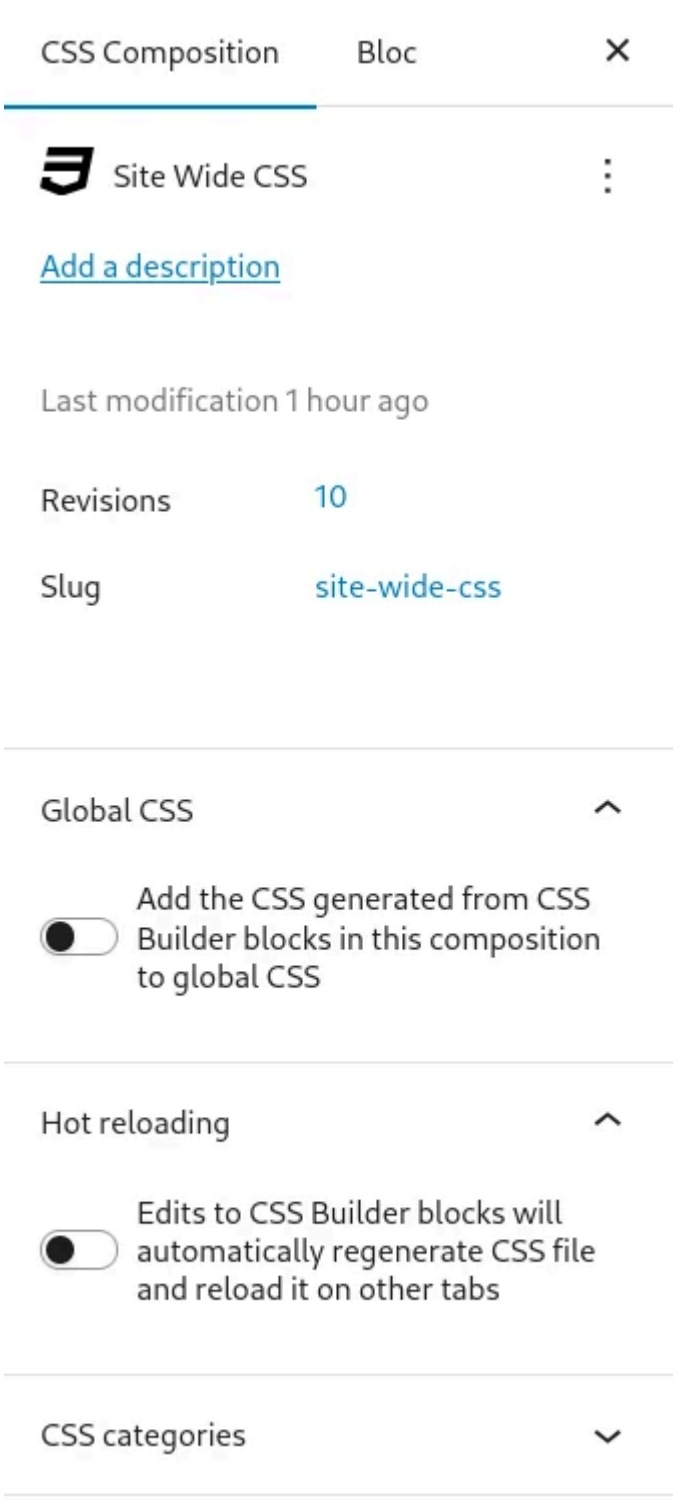


Changing some styles of elements of the EDD login form...



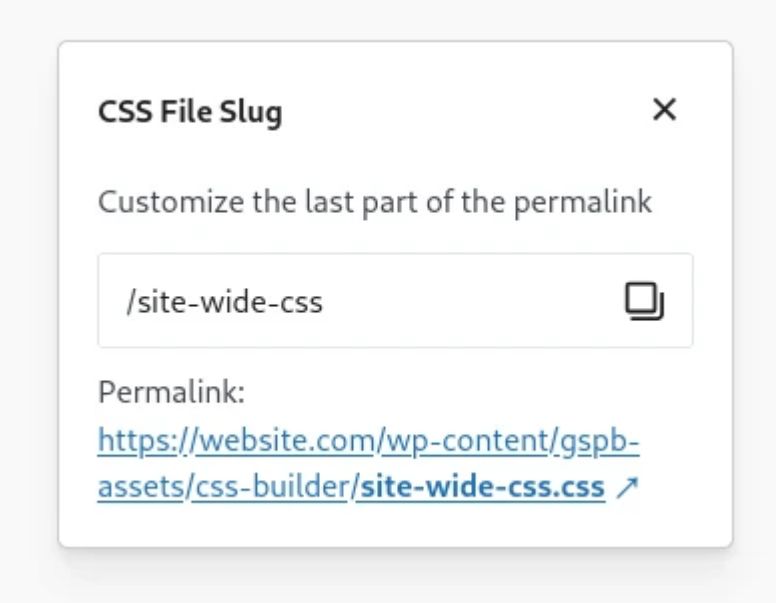
Some components that may be used everywhere... (there's only one CSS Builder block here as you can see, but multiple classes created)

When you click on `Composition` tab, you get access to some options:

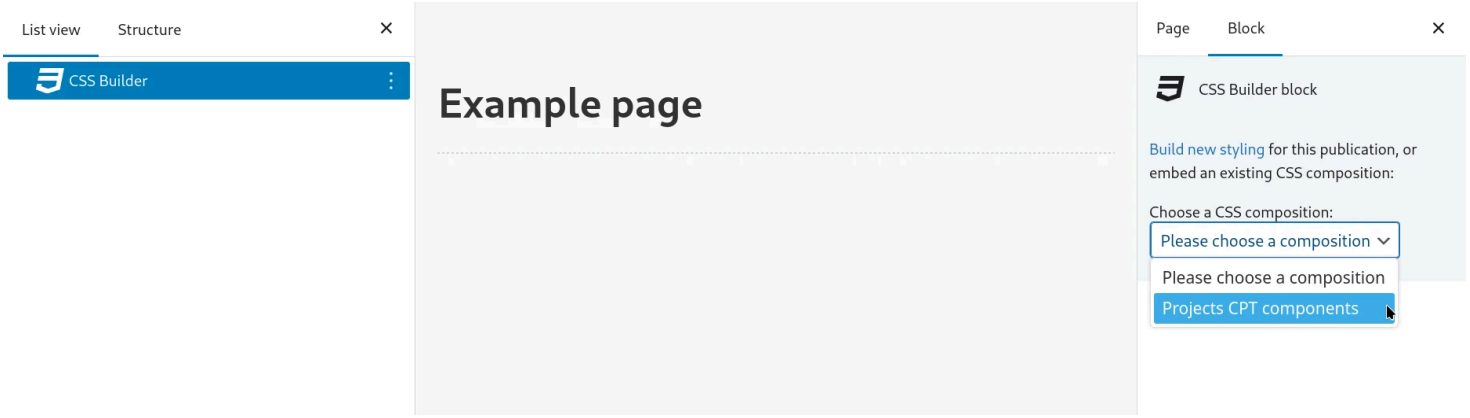


As this creates a CSS file, it is easy to implement hot reloading, other tabs just have to query the new stylesheet via javascript

You can change the CSS file name:

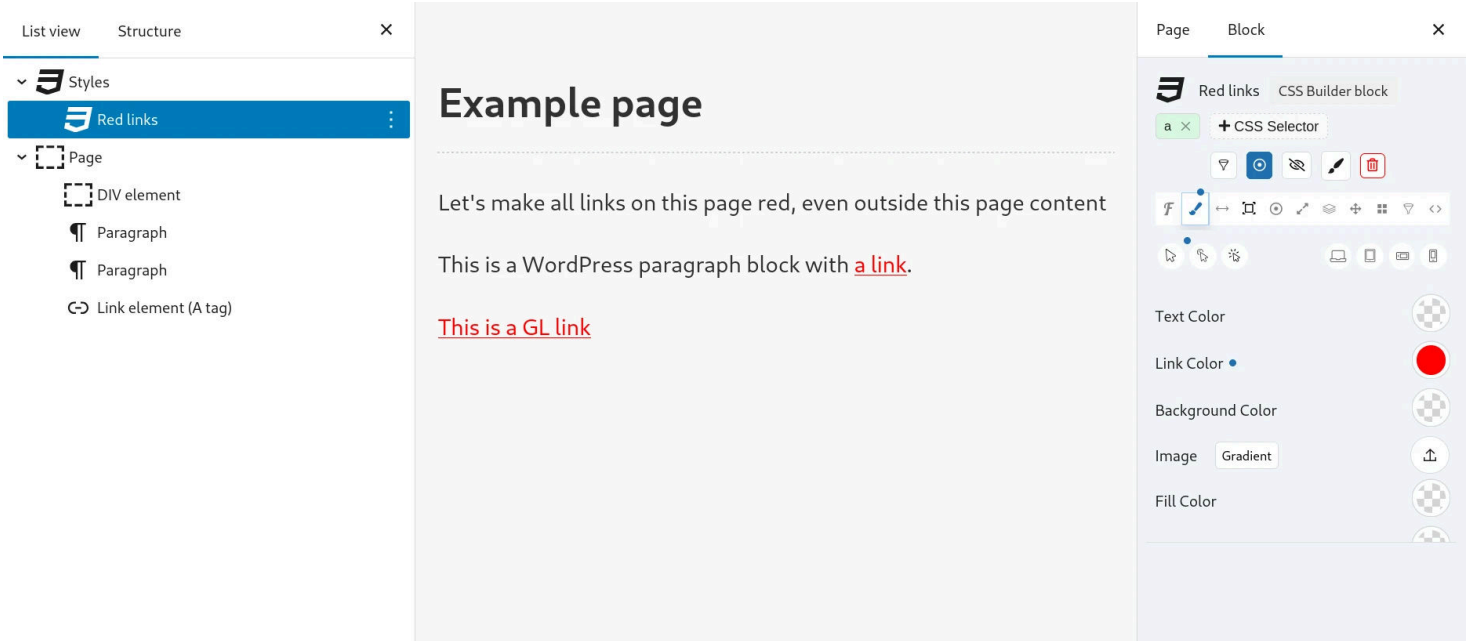


Then, to facilitate embedding CSS Compositions (not globals) into other post types, this is what you get when adding a CSS Builder block:



You may either start designing new selectors, or directly embed existing compositions.

Let's try starting new, and change the styling of links (`a` tags) for this page:



Some advantages

- Foundation for shareable designs, either Greenshift or anything. Marketplace of design
  Example: build nice components, layouts etc with Greenshift, sell your generated CSS. They can be used anywhere for any web platform
  Selling your design compositions is also an option if your clients also use Greenshift
- Self and visual documenting of css / designs
- Easily switch between multiple designs with same layout (client showcase for example) just by selecting an other composition in CSS Builder
- Use Greenshift and WordPress to build/version CSS for any usage
- Only a CSS file after generation, so it's efficient to load
- Less code in publications in comparison to global classes: faster page generation and lighter on database (especially with many revisions of publications)

  Global classes import all their attributes in the publication, each time they are used.
  Here is a comparison with classic classes, just only by styling color:reddish. This can grow up very fast with complex designs

```
<!-- wp:greenshift-blocks/element {"id":"gsbp-21c42dc","dynamicGClasses":[{"value":"red-
p","type":"global","label":"red-p","css":"body .red-p:is(p){color:#c70c0c;}","attributes":{"styleAttributes":
{"color":["#c70c0c"]}},"selectors":[],"originalID":"gsbp-c7ce1c5","originalBlock":"greenshift-blocks/
element"}],"textContent":"This is some text with global class","tag":"p","className":"red-
p","localId":"gsbp-21c42dc"} -->
<p class="red-p">This is some text with global class</p>
<!-- /wp:greenshift-blocks/element -->


<!-- wp:greenshift-blocks/element {"id":"gsbp-3c16719","textContent":"This is some text with classic
class","tag":"p","className":"red-p-classic","localId":"gsbp-3c16719"} -->
<p class="red-p-classic">This is some text with classic class</p>
<!-- /wp:greenshift-blocks/element -->
```

Please share your thoughts on the Facebook Greenshift Group, I may fine tune this document, and upvote the linked feature request on Greenshift website!
Thanks